# Meeting with Schneider Electric – Frank

* Questions for Frank
  + Missing basic instructions like MOV and JSR?
    - We can’t write a MAIN Ladder file and JSR to specific files on startup. Does it happen automatically?
  + How do we retrieve the barcode data?
* Configuring Modules
  + Double click to configure.
  + Easiest way to configure is using topological addressing.
  + Elementary Variables
    - Name
    - Type
      * EBOOL
    - Address
      * %I0.1.0
      * %I0.1.1
      * We don’t have to skip to the 8th, 16th, etc. These are all on the same channels.
* Sensor
  + Detect heat? Turn on.
* Sections
  + Solve top to bottom, not simultaneously.
  + MAIN
  + SR Sections (Subroutine)
    - Use C coils to call subroutines from your primary sections.
* Arrays
  + Elementary Variables
* Variables
  + Everything is global! Elementary Variables.
* Syntax
  + = for comparison.
  + =: for assignment.
* Derived Function Blocks
  + Name
  + Inputs
    - Ex. Bin1, Bin2, BinN
  + Outputs
    - Ex. Sensor\_Light1 … n
  + I/O
    - Whats\_In\_Bin
    - Whats\_in\_Lock
  + Create sections for our logic.
    - Whats\_In\_Bin
    - Whats\_In\_Lock
    - Get\_Barcode\_Scan
    - Overall\_Control
    - Sound\_Horn
* Modes
  + Standard Mode
    - Connect and program the controller.
  + Simulation Mode
    - Analyze Project.
    - Rebuild Project.
    - Build Code
      * MEM
      * BUILT
    - Green Question Mark (Simulator)
    - Transfer Project to PLC
    - Run Project? Yes.
    - Animation Tables
      * We can see what’s happening in memory.
* Rebuild
  + Stops the PLC to load the program on.
  + From here on we only have to build to make changes to the PLC. We don’t have to stop it anymore.
* HMI
  + Barcode Scanner
    - One barcode scanner per work station.
  + We can assign addresses to variables.
    - Then, the screen (Magellus) will be able to access that information.
* Project Design
  + Two Function Block Section
    - Main FBD
    - Barscan needs to be its own because we’re reading from Serial.
      * Scheduled to read in order (1 … 4)
      * READ\_VAR Block
        + %MW – Memory Word
        + Modbus serial compatible.
        + Will take a Modbus call.

A protocol call using a Modbus protocol.

* + - * + Reads memory words inside of the barcode scanner.
        + When the barcode scanner sees the barcode it will place the physical barcode number into one or multiple Modbus registers.
        + What does the READ\_VAR do?

The scanner is always scanning or on button press.

Constantly pulling the scanner until we see a non-zero number in the registers.

* + - * + IF non-zero? Do something!

How many registers are we going to use?

* + - * + NUM = first Modbus register we’re reading from.

We have to find out which register the barcode ends up in and assign it here.

Ex. Registers 1 – 5

NUM = 1 (Start)

NB = 4

Probably 4.

* + - * + GEST

Read\_var\_gest

* + - * ADDM Block
        + Accepts a string.
        + EN and ENO

Conditionally enable the block from solving.

OUT 🡪 READ\_VAR: ADR

* + - * + Read from routing address.
        + Probably 0.0.0.1
        + Rack, Slot, Channel, Slave #

0 – 4 for each scanner.

* + - * + **Allow Dynamic Arrays (Settings 🡪 Variables)**
        + READ\_VAR now knows where to send the modbus message.
        + Reading from one on the controller.
        + For length of 4 (NB).
    - 4 Derived Function Blocks
      * Create sections for our logic.
        + Whats\_In\_Bin
        + Whats\_In\_Lock
        + Get\_barcode\_Scan
        + barcode\_Scan (LD)

Modbus reading between PLC and scanner.

* + - * + Overall\_Control
        + Sound\_Horn
  + NOM Module
    - Configure as 0.2; 0.1 is the old module.
    - Function: Modbus Link
    - Type: Master
    - RS 232 or RS 485.
    - RTU: Number of stop bits.
    - Parity: Even, Odd, None.
    - **Has to match the barcode scanner**.
  + Help
    - Right click Function Block, Help on Type
      * Documentation
    - Ex. Example of Reading Words via Serial Link of Modicon M340 Processors
  + Procedure
    - Main\_FBD solves.
      * Calls Workstation\_Code\_#
        + Passes what’s in bin, what’s in locks.
        + Executes code.

(You can right click, refine, to see what’s happening in a particular instance).

* Exporting Files
  + ZEF, XEF
    - Full Application
    - Always have a backup!
* Importing Files
  + Brian’s DB
    - Import Excel \*.txt file.